Bottleneck Analysis for Routing and Call Scheduling in Multi-hop Wireless Networks

Cristiana Gomes, Stéphane Pérennes, Hervé Rivano

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Joint Gathering and Coloring problem in Wireless Mesh Networks

Cristiana Gomes — Stéphanne Pérennes — Hervé Rivano

N° ????

Mai 2008

Thème ?

Rapport de recherche

N 0249-6399

Joint Gathering and Coloring problem in Wireless Mesh Networks

Cristiana Gomes, Stéphanne Pérennes , Hervé Rivano

Thème ? —
Projets Mascotte

Rapport de recherche n° ???? — Mai 2008 — 11 pages

Abstract: We deal with a bandwidth allocation problem as a Joint Gathering and Coloring problem. We have to find sets of compatible communication links in a network, called *rounds*, such that the routers requirements of bandwidth are achieved.

We present a mixed integer linear programming model and we use a branch-and-price algorithm to solve the problem. A column generation approach is used to avoid dealing with the exponential set of rounds and the branch-and-bound algorithm to turn integer the flow (network packets). We run experiments on networks from the literature, with different number of gateways.

Experimental results as well as theoretical insights let us conjecture that gateway neighbourhood analysis is enough to determine the problem solution in the cases where the largest bottleneck is around the gateway. The *round-up property* seems to hold for our problem.

Key-words: Wireless Mesh Networks, coloring, routing, multi-hop, Branch and Price algorithm

Problèmes du routage et de coloration dans les réseaux maillés sans fil

Résumé : Nous nous intéressons à un problème d'allocation de bande passante que peut-etre considé comme le problème de *Joint Gathering and Coloring.* Nous devons trouver dans un réseau un ensemble de liens de communications compatibles appelés *rounds*, tels que les demandes en bande passante des routeurs soient satisfaites.

Nous présentons un modèle de programmation linéaire mixte et nous utilisons un algorithme de *Branch and price* pour résoudre le problème. Une approche par génération de colonne est utilisée afin d'éviter de considerer le nombre exponentiel de *rounds* et un algorithme de *Branch and bound* rend le flot entier (paquets). Nous faisons des tests sur des réseaux issus de la littérature, avec plusieurs nombre de *gateways*.

Nous résultats expérimentaux ainsi que des arguments théoriques nous permettent de définir comme conjecture le fait que l'analyse du voisinnage du *gateway* est suffisante afin de résoudre le problème dans les cas où le plus grand goulot se trouve autour du *gateway.* La propriété d'arrondi supérieur (*round-up*) semble être valide pour notre problème.

Mots-clés : Réseaux maillés sans fil, coloration, routage, multi-saut, Branch and Price algorithm

1 Introduction

In wireless networks, the communication channels are shared by the terminals. Thus, one of the major problems faced is the reduction of capacity due to interferences caused by simultaneous transmissions [1]. In this work, we call a *round* a collection of links that can be simultaneously activated in the network.

We deal with a bandwidth allocation problem called *Round Weighting Problem (RWP)* [2], that jointly considers the *multi-commodity flow problem* and the *weighted fractional coloring problem.* We have to find sets of compatible communication links in a network, called *rounds*, such that the routers bandwidth requirements are achieved.

A network topology is a *communication graph* G where the nodes are the routers and the edges are the links. Interferences between links are given as a *conflict graph* G_c. We consider the bandwidth problem as a flow problem. Therefore, the flows in the edges represent the badwidth. We deal with the case where data are sent to the gateways (gathering), therefore the flow subproblem is a single-commodity problem. We call this problem Single-Commodity RWP or RWP_{SC}. It can be applicable to a wide range of problems in multihop wireless networks.

The RWP_{SC} input corresponds to G, G_c, and the network bandwidth proportion between each router and a given set of gateways. Each edge e of G receives a positive value $b(e)$ that represents the flow problem solution. Simultaneously, we find a set of *rounds* $\mathcal{R}$ with weights $w(R_i)$ achieving the routers bandwidth ($\sum_i w(R_i) \geq b(e) : e \in R_i$), such that the total weight ($\sum_i w(R_i)$) is minimized.

We consider the interference distance d is the same to all links. The conflict graph G_c is the line graph power d, $L^d(G)$. For example, if $d = 2$ a *round* is an *independent set* of G_c or an *induced matching* in G. In this case, the RWP_{SC} is a *strong edge-coloring* (or a *L(1,1) edge labeling*) of the multigraph G. It is a multigraph because the edges have weights defined by the flow going through them. These weights are simultaneously defined representing the best routing possible to reach the minimum number of colors W. So, this problem can be seen as a *joint Gathering and Coloring problem.*

We present a mixed integer linear programming model and we use a branch-and-price [3] algorithm to solve the problem. A column generation [4] approach is used to avoid dealing with the exponential set of rounds and the branch-and-bound algorithm to turn integer the flow (network packets). We run experiments on networks from the literature, with different number of gateways. Experimental results as well as theoretical insights let us conjecture that gateway neighbourhood analysis is enough to determine the problem solution in the cases where the largest bottleneck is around the gateway. The *round-up property* seems to hold for our problem.

The model in this article can be useful as a benchmark for networks with distributed links scheduling, like in IEEE 802.11. It can also be useful in a context where centralized links scheduling can be adopted, like in IEEE 802.16, that can directly take advantage of our analysis.

This paper is organized as follows. In the next section, we discuss the related works. In section 3 we define the problem. In section 4, we give a description of the Branch and Price algorithm adopted. Experimental results and analysis about gateway neighbourhood importance and round-up property are presented in section 5. We conclude the paper and give the future directions in section 6.

2 Related Works

A key issue in wireless networks is the interferences produced between neighboring routers. It is important to know what are the sets of transmission links that can be active at the same time, the rounds. The Round Weighting Problem *RWP* was treated in [2]. The authors make a dual analysis and propose approximation algorithms for some specific network topologies. They show the NP-hardness of this problem by proving that the well-known NP-hard problem of finding the *Fractional Coloring* on unit graphs reduces to it. The Fractional Coloring was proved NP-hard by [5].

The problem *SP-RWP* was treated in [6], the authors give exact bounds for the problem on grids. They mainly prove that the throughput is determined by the bottleneck around the base station.

An algorithm enumerating a tractably large subset of simultaneous transmission rounds has been developed in order to compute an approximated solution for maximum throughput using linear programming (LP) in [7]. Solving the full LP problem means generating an exponential set of scenarios which is intractable even for small networks as seen in [8]. To cope with this issue, column generation methods have been introduced, e.g. [4], [4], [9] and [10]. In [4], a multi-objective study about the Round Weighting Problem was presented. To solve integer programs with a huge number of variables, the Branch-and-Price (BnP) [11] combines Branch-and-bound with

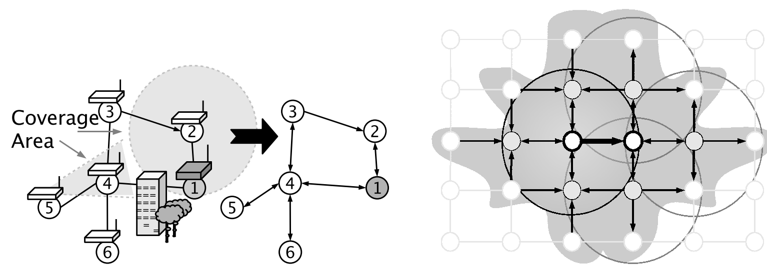
Column Generation. In a network the flow should be integer to better represent packets units, consequently a branch and price method was used [3].

In [12], a similar problem, the *Round Scheduling Problem* was treated. The relation with the round weighting problem is the following: if one must repeat rounds scheduling many times then the problem is equivalent to the *RWP*. The authors prove lower bounds on the number of rounds required for any two-dimensional grid and describe protocols for n x n grids with n odd that are optimal. Note that, in [12] the distance of interference is $d_I > 1$ and it is not symmetric because they deal with the exact case of gathering (directed interference).

The capacity of radio networks has been extensively investigated. Under specific routing, interference and probabilistic traffic assumptions, it has been shown that a wireless network cannot provide better than a throughput of order $\theta(W/\sqrt{n})$ bps to each node [13] and [14].

3 Hypotheses and problem definition

In this section we give some definitions that will help to understand the problem. The RWP_{SC} can be modeled as a graph problem. A wireless topology is represented as a digraph $G = (V, E)$, the *communication graph* as illustrated on figure 1(a).



(a) Communication Graph (b) Edge Collision Area

Figure 1: Network modeling

The set of routers is represented by the set of nodes V. The set of edges $E \subseteq V \times V$ corresponds to the communication links from the real network. If a router j is located within the transmission range of a router i, considering range distance, obstacles, directional antennas, etc, then there is an edge $(i, j) \in E$.

A round can be simply defined by any set of edges pairwise at distance at least $d_I^s + 1$. It defines a symmetric interference model that permits the calls happen in both directions (download or upload). The most basic case is $d_I^s = 0$, where the set of rounds $\mathcal{R}$ is simply the set of the matchings of G. We solve the case with $d_I^s = 1$ but our method accept any d_I^s. In other words, an active link (i, j) interferes with another links located within the interference range defined by d_I^s of the router i. The set $I_{u,v}$ is composed by all links interfering with the link (u, v) that defines the *collision area*. In figure 1(b), we can see an example in a grid graph. Flexible binary interference models can be adopted. We consider a symmetrical collision area in a way to permit communication in both directions. That is all edges in bold on figure 1(b) will be interfered in both directions.

In the *conflict graph* G_c, the nodes represent the communication links of G and the edges connect two incompatible nodes defined in $I_{u,v}$. So a *round* can be seen as an *independent set* of G_c or an *induced matching* in G. The RWP_{SC} is in fact a *strong edge-coloring* (or a *L(1,1) edge labeling*) of the multigraph G with weights (flow). These weights are simultaneusly defined representing the best routing possible to reach the minimum number of colors.

A round in a wireless network corresponds to a set of links that can be active at the same time without making interferences between them. The size of the complete set of rounds is exponential. We consider a column generation approach to select as required the rounds to improve the solution of the problem. The round definition guarantees that the communication will be multiaccess interferences free.

The bandwidth should be allocated between the set of nodes V_r and the set of gateways V_g ($V_r \cup V_g = V$ and $V_r \cap V_g = \emptyset$). The RWP_{SC} input corresponds to the communication graph $G(V_r \cup V_g, E)$, the conflict graph G_c, and the network bandwidth proportion to each router. In the output, each edge $e \in E$ receives a positive integer value $b(e)$ that represents the bandwidth from the routing problem solution. Simultaneously to the routing problem, we have to find a set of rounds with their weights $w(R_i)$ achieving the routers bandwidth

$(\sum_i w(R_i) \geq b(e) : e \in R_i)$, such that the total weight $W = \sum_i w(R_i)$ is minimized. From this set of rounds can be deduced the paths followed by the data.

4 Branch and Price Approach

We use a Branch-and-Price (BnP) algorithm to turn integer the flow variable $x^v_{i,j}$, in order to better represent the packets in a network. The BnP combines Branch-and-bound (BnB) with Column Generation to solve integer program with a large number of variables. Each node of the BnB tree corresponds a linear program to be solved, a *constraint stack* that should be considered in this linear program, a *list of variables* that should be integer and a position in the *execution stack*. Each time, the *execution stack* saves only a part of the Branch-and-Bound tree. As we use a Depth-First approach, these parts corresponds to exactly paths that we follow deep into the tree. We stop when we find the best objective, otherwise, a backtracking is needed.

At each node of the tree, we solve a linear relaxation of the SC-RWP problem with column generation (see subsection 4.1). Therefore, each node is repeated until no further variables price out favorably. With the optimal solution of the column generation we have a *list of variables* and we can choose among them how to bound the children nodes. We choose a not integer variable to branch from the list of variables. We respect a priority order, the first variables corresponds to the flow variables around the gateway and the next ones are chosen randomly. We explain in the section 5 why this priority order was adopted.

We start with a relaxed model at the root node with an empty *constraint stack*. For the execution of each node of the Branch-and-Bound tree we create two children with a copy of the father *constraints stack* and a new constraint each one. We push a child node of the BnB tree on the execution stack. This node is then on top of the *execution stack* and it will be executed next.

If at any point the relaxed model of a BnB node becomes infeasible, this node is pruned. That is it is popped out from *the execution stack*.

The best node is a BnB node with all its variables integer and the objective is better than the current best solution. So, this node will be used as current best solution. Otherwise, we continue splitting the problem in one or two new problems branching on a variable of the *list of variables* that is not integer yet. The figure 2 shows an example of a part of the BnB tree of the BnP algorithm.

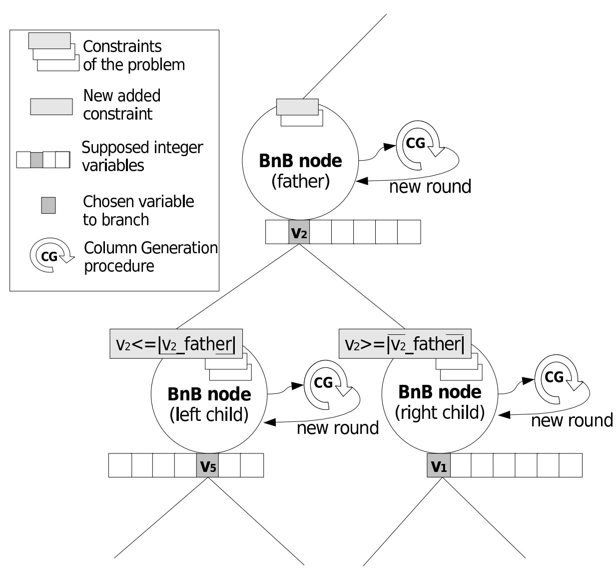


Figure 2: The Branch and Price algorithm

Actually, we use just a very simple algorithm, which is sufficient since this problem have good characteristics, as we will see in the section 5. These characteristics turn easier the problem and the solution is found efficiently because we can cut very early the BnB tree, it will be explained in section 5.

4.1 Column Generation Method

The problems considered are the RWP+*MinMaxLoad* and the RWP+*MinTime* taking into account the complete set of rounds. As the number of rounds is exponential, the number of columns of the constraint matrix is exponential. The key idea of the column generation is that it is not needed to list explicitly all of the columns of the problem formulation, but rather to generate them only "as required" [15]. The problem is decomposed into a master problem and a sub-problem. We solve the master problem with a small subset of columns, which serves as an initial basis. The sub-problem is then solved to check the optimality of the solution under the current master basis and to generate new columns for the master problem. This procedure repeats until the master problem contains all columns necessary to find the optimal solution of the original problem. Each column corresponds to one round.

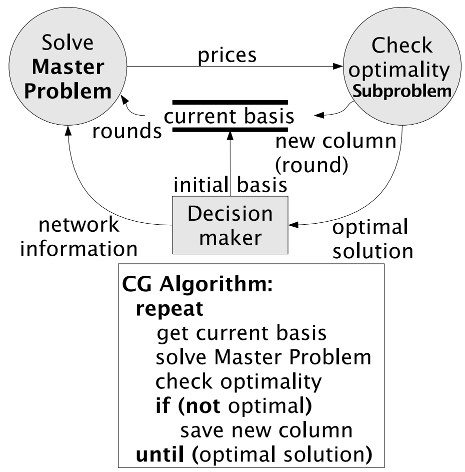


Figure 3: Column generation algorithm and data flow diagram.

In each iteration, if the sub-problem can find a new column that may improve the master solution, this column is inserted in the master basis and a new master solution is computed. If the sub-problem cannot find a new column, it means that the solution of the problem is optimal. This algorithm is represented on figure 3. This column generation approach is close to the one proposed by Gilmore and Gomory [16]. The notation and the decomposition of the problem in master model and sub-model are based in [9]. We adapted the master model to the WMNs context. this model was presented in [4].

4.1.1 Master problem formulation

We define the following variables: Let the variable $x^v_{i,j}$ denotes the flow from the router v over link i, j. The demand from each router v is represented by the parameter b_v. Let the binary parameter $a^r_{i,j}$ be 1 if link (i, j) is active in the round r, and 0 otherwise.

Recall that set $I_{u,v}$ is composed by all links interfering with (u, v). We define $F^{(i,j)}_{(u,v)} = 0$ if $(i, j) \in I_{u,v}$ and 1, otherwise. We define w_r as the fraction of time that round $r \in R$ is scheduled. Consequently, there is an induced edges capacity $c_{i,j} = \sum_{r \in R} a^r_{i,j} w_r, \forall\, (i, j) \in E$.

The master problem can be defined as follow: Given a graph $G(V_r \cup V_g, E)$, a set of routers demand b_v with $v \in V_r$ and a set of rounds R, the problem is to assign a weight w_r to each round $r \in R$. The weights represent the amount of time a round will be activated. The total amount of time needed to satisfy all demand will be $w_R = \sum_{r \in R} w(r)$. From the edges of the rounds can be deduced the paths followed by the data as illustrated in figure 4. It may happen that some of the rounds r have a weight equal to zero. The load in each router $i \in V_r$ is given by $l_i = \sum_{v \in V_r} \sum_{j \in V/(i,j) \in E} x^v_{i,j}$.

The constraints of the master problem expressed as a linear programming model are the following:

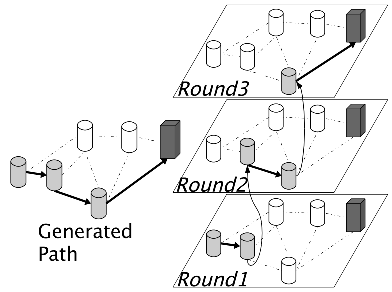


Figure 4: Path deduction from a set of rounds.

$$\sum_{i \in V/(v,i) \in E} x^v_{v,i} = b_v, \forall v \in V_r \tag{1}$$

$$\sum_{j \in V_g} \sum_{i \in V_r/(i,j) \in E} x^v_{i,j} = b_v, \forall v \in V_r \tag{2}$$

$$\sum_{i \in V_r/(i,j) \in E} x^v_{i,j} - \sum_{k \in V/(j,k) \in E} x^v_{j,k} = 0, \forall j, v \in V_r \tag{3}$$

$$\sum_{r \in R} a^r_{i,j}.w_r - \sum_{v \in V_r} x^v_{i,j} \geqslant 0, \forall i, j \in E \tag{4}$$

Constraints (1-3) correspond to the flow constraints. Constraints (1) define the flow leaving its source router and constraints (2) define the flow arriving in a gateway. Constraints (3) represent the flow conservation, that is, the flow entering an intermediate router equals the flow leaving that router. Constraints (4) assign weights to the rounds to satisfy the flow in the edges.

4.1.2 Sub-problem formulation

The sub-problem generates a round r with the minimal *reduced cost* $\left(1 - \sum_{(i,j) \in E} p_{(i,j)}.a^r_{i,j}\right)$ to enter the master basis. To express the sub-problem as a linear programming model, we have to define some additional notations. Let the parameter $p_{(i,j)}$ be given by the dual variable associated with the constraints (4) of the master problem. Consider the binary variable $u_{(i,j)} = 1$ indicating if the edge (i, j) enters the round to be added to R, $u_{(i,j)} = 0$ otherwise. The sub-problem can be seen as the *Maximum Weighted Independent Set Problem* which is NP-hard [17]. The parameter $p_{(i,j)}$ corresponds to the weight of the edges. The objective function is to maximize the sum of the weights of all active edges respecting interferences.

The formulation of the sub-problem is the following:

$$\max \sum_{(i,j) \in E} p_{(i,j)} u_{(i,j)} \tag{5}$$

$$u_{(i,j)} + u_{(k,l)} \leqslant 1 + F^{(k,l)}_{(i,j)}, \forall (i,j) \in E, \forall (k,l) \in E \tag{6}$$

The objective function (5) searches the maximum weight, which is equivalent to the minimum reduced cost. The parameter $p_{(i,j)}$ guides the column generation to select the best round. Constraints (6) avoid interferences according to the interference model in F.

If the value of the objective function in the sub-problem is strictly greater than 1 (e.g. the reduced cost is negative), a new column $u_{(i,j)}$ is found and the master basis is expanded. Otherwise, the master problem already gives the optimal solution to the original problem.

5 Result Analysis

The model is implemented using the AMPL modeling language and the Branch-and-Price algorithm is implemented using ILOG Concert Technology. The instances are solved using the commercial software Cplex

Table 1: Networks topologies and results

Network	Gateways	Nodes	Edges	$W_f(b_v = 1)$	$W_i(b_v = 1)$
pdh	1	11	34	16	16
pdh	2	11	34	9.5	10
polska	1	12	18	15	15
atlanta	1	15	22	17.666	18
atlanta	3	15	22	7.71428	8
newyork	1	16	49	18.5	19
newyork	3	16	49	6.6666	7
france	1	25	45	54	54
france	3	25	45	14.5	15
nobel	1	28	41	38	38
giul	1	39	172	49	49

version 10, on a desktop PC with one gigabyte of RAM. We use some mesh networks topologies (only the graph representation) from [18].

We consider the bandwidth problem as a flow problem. Therefore, the flows in the edges represent the badwidth. In the table 1 are given the networks topologies characteristics. The solutions W_i to RWP_{SC} with integer flow obtained by the BnP algorithm are shown. The table also shows the solutions W_f to RWP_{SC} with fractional flow obtained using only Column Generation algorithm. We consider $b_v = 1$. The computation time to solve any instance was low, of the order of tenths of seconds.

In the table 1, we define a simple interference model where each edge interferes with another one if the distance between them in graph G is lower than 2, but the model accepts any binary interference model. We consider equal bandwidth requirements b_v (uniform bandwidth) for all routers, which can be interesting for fair systems. We can see in the table 1 that if there are 2 or more gateways, W is not exactly divided by the number of gateways because they have different absorption rates or because they are close and their critical regions make interference one with the other. As $b_v = 1$ the BnP algorithm give us a mono-routing that can be interesting to avoid dealing with the packet-reordering problem.

The Integer Round-Up Property (IRUP) seems to hold for the RWP_{SC} in our tests resuts. The table 1 shows that $W_i = \lceil W_f \rceil$. It is due to the fact that there is a huge concentration of traffic around the gateway (critical region) because all the flow goes toward it. That is why we give priority to these variables in the BnB algorithm 4.

If we consider the interference distance d_i is the same to all links. The conflict graph is the line graph power d_i, $L^{d_i}(G)$, where G is the topology graph. If each router sends the same b_v units of flow during W slots of time, there is $n.b_v$ units of flow on the edges around the gateway. It should be the most critical point due to the flow concentration. If we transform these flows (weights) in new nodes of the conflict graph G_c, we can see large cliques composed of the gateway and its neighborhood. The number of colors to cover the largest one is a lower bound to our problem solution. It is known that $\omega(G_c) \leqslant \chi_f(G_c) \leqslant \chi(G_c)$ for any graph G_c, where $\omega(G_c)$ is the clique number of G_c (with weights). We observe that the clique number represents already the solution of our problem as shown in figure 5, the letters represent the rounds.

In our analysis, it is the case in all network graphs that have no bottleneck worse than this one formed around the gateway and the flow can arrive in this region without problems, what usually happens in real backhaul networks. Otherwise, the quantity of time depends of the size of the cliques formed in other parts of the network. In [4] was presented an algorithm that can be used to find bottlenecks position.

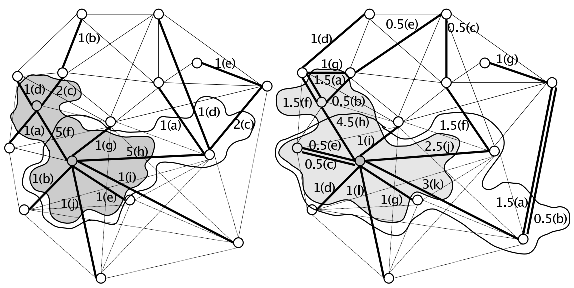
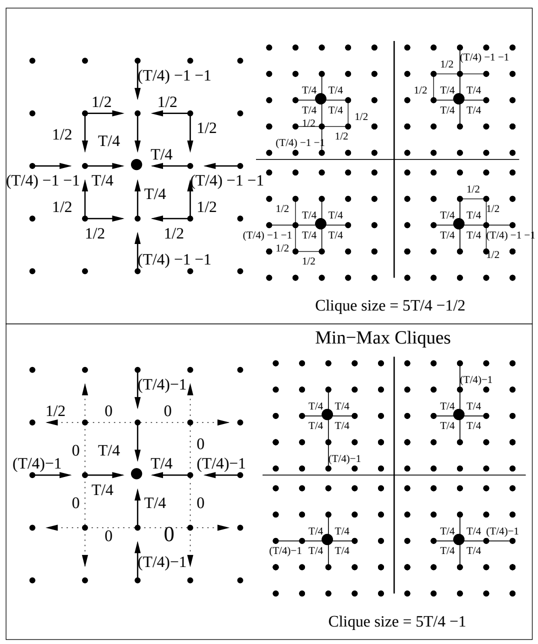
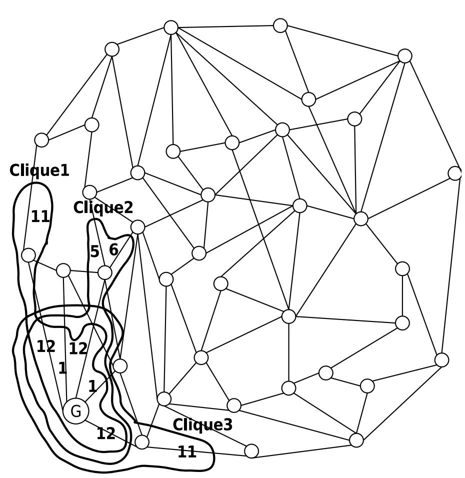


Figure 5: NewYork Network ($T = 15$) with solution $W_f = 18.5$ and $W_i = 19$



(a) Grid Networks, solution $W = 5.\frac{T}{4} - 1$



(b) Giul Network, $T = 38$ and solution $W = 49$

The RWP_{SC} can be easily solved once we have covered the critical region. The links out of this region have *slacks* of activation. An edge have a *slack* when it has several possible options to get activated forming a *round* with edges on the *critical region*. It is easy to assign time slots (colors) to the edges out of this region because once we have covered the critical region they will have several possibilities to get activated whitout change the total routing time W.

The traffic around the gateway is well distributed between the cliques because the model try to minimize the maximum clique, it results in cliques of the same size for the fractional result and almost the same size for the integer result. This explains the integer colouring is simply the round up of the integer colouring ($W_u = \lceil W_s \rceil$). It allows to cut efficiently the BnB tree in the BnB algorithm 4 putting the current best solution objective value equal to $\lceil W_s \rceil + 1$.

We notice that the colors (time slots) used in the weighted min-max cliques around the gateways is enough to color all routes. The weighted min-max cliques around the gateway can be determined by a distribution of the total flow comming from the nodes over its links. The cliques are generated in a way to minimize the maximal one, consequently it will minimize the total number of colors needed (total routing time W). The figure 6.a shows an example of a bad traffic distribution giving a worse result and the best configuration for grid networks with the gateway in the center. The best configuration shows that there is 4 nodes close to the gateway that have to go far from it and come back after through the central cliques. This routing minimizes the maximal cliques. The Grid result in figure 6.a has been proved in [6]. Another result can be seen in figure 6.b, we show a larger network and the maximum clique with weight equals the optimal value.

In our tests analysis, the path followed by the routers is close to the *Minimum Spanning Tree (MST)* but with the flows respecting the flow distribution around the gateway (min-max weighted cliques). We suspect that we can give a good routing considering the constant λ that is a interference cost by link. The constant $\lambda(i, j)$ is the quantity of links (i, j) interferes, that is $|I_{u,v}|$. It makes sense because all routers are source and send the same quantity of flow on an unique path, so these paths interfere less between themselves than others. There is an example in the figure 7 that shows the balanced MST. We can also see, the complete routing with integer flows forming three cliques around of the gateway with the weights 18, 18 and 17.

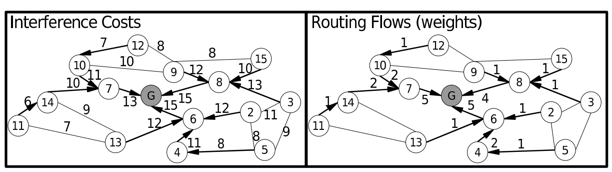


Figure 7: MST backbone of the Atlanta network

6 Conclusion and perspectives

In this article, we solve the Round Weighting Problem in order to satisfy a given demand subjected to the multiaccess interferences. We define a mixed integer linear programming model and a branch-and-price approach to solve the problem. A column generation algorithm is used to avoid dealing with the whole rounds set that is exponential and the branch-and-bound algorithm to turn the flow integer (network packets). We make experiments with networks from the literature with different number of gateways.

The RWP_{SC} can be easily solved once we have covered the critical region and the *round-up property* holds for the SC-RWP in our tests instances. We are getting further the proof of the problem having or not the *round-up property* in specific class of graphs, eg. planar graphs. At first, we need to define the algorithm that gives a lower bound (the clique number) to our problem and prove for which class of graphs it is tight. Our future step will be to deal with dynamic traffic and minimizing the backbone reconfiguration.

Acknowledgments

The authors wish to thank Jean-claude Bermond and Joseph Yu for their helpful comments. C. Gomes is funded by CAPES, Brazil. This work has been partially funded by European project IST/FET AEOLUS, ANR-JC OSERA, and ARC CARMA.

References

[1] W. Wang, X. Li, O. Frieder, Y. Wang, and W. Song, "Efficient interference-aware TDMA link scheduling for static wireless networks," in *International Conference on Mobile Computing and Networking (Mobicom)*. ACM Press, 2006, pp. 262–273.

[2] R. Klasing, N. Morales, and S. Pérennes, "On the complexity of bandwidth allocation in radio networks," *Theoretical Computer Science*, 2008, to appear. [Online]. Available: http://www.elsevier.com/locate/tcs

[3] H. R. Cristiana Gomes, Gurvan Huiban, "A branch-and-price approach to the bandwidth allocation problem in wireless networks," in *International Symposium on Combinatorial Optimization (CO 2008)*, University of Warwick, Coventry, UK, March 2008, abstract.

[4] C. Gomes and G. Huiban, "Multiobjective analysis in wireless mesh networks," in *15th Annual Meeting of the IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2007.

[5] B. N. Clark, C. J. Colbourn, and D. S. Johnson, "Unit disk graphs," *Discrete Math.*, vol. 86, no. 1-3, pp. 165–177, 1990.

[6] C. Gomes, S. Perennes, P. Reyes, and H. Rivano, "Bandwidth allocation in radio grid networks," May 2008.

[7] S. Mukherjee and H. Viswanathan, "Throughput-range tradeoff of wireless mesh backhaul networks," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 3, pp. 593–602, 2006.

[8] P. R. Cristiana Gomes, Christelle Molle, "Optimal design of wireless mesh networks," in *9èmes Journées Doctorales en Informatique et Réseaux (JDIR)*, 2008.

[9] J. Zhang, H. Wu, Q. Zhang, and B. Li, "Joint routing and scheduling in multi-radio multi-channel multi-hop wireless networks," in *International Conference on Broadband Networks (Broadnets)*, vol. 1. IEEE Press, October 2005, pp. 631–640.

[10] P. Värbrand, D. Yuan, and P. Björklund, "Resource optimization of spatial tdma in ad hoc radio networks: a column generation approach," in *INFOCOM*, vol. 2. IEEE, March-April 2003, pp. 814–824.

[11] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance, "Branch-and-price: Column generation for solving huge integer programs," *Oper. Res.*, vol. 46, no. 3, pp. 316–329, 1998.

[12] J.-C. Bermond and J. Peters, "Efficient gathering in radio grids with interference," in *Septièmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications (AlgoTel'05)*, Presqu'île de Giens, May 2005, pp. 103–106.

[13] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, March 2000.

[14] G. Méheut, S. Pérennes, and H. Rivano, "Evaluation stochastique et simulation des réseaux radio," INRIA, Research report 5989, September 2006. [Online]. Available: http://hal.inria.fr/inria-00102039

[15] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network flows: theory, algorithms, and applications*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2003.

[16] P. Gilmore and R. Gomory, "A linear programming approach to the cutting stock problem," *Operations Research*, vol. 9, pp. 849–859, 1961.

[17] M. Garey and D. Johnson, *Computers and intractability*. New York: W. H. Freeman and Company, 1979.

[18] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly, "SNDlib 1.0-Survivable Network Design Library," in *Proceedings of the Third International Network Optimization Conference (INOC 2007), Belgium*, April 2007.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399